Controlling the Spindle Speed when Milling Free-Form Surfaces using Ball-End Milling Cutter

Abdulwahab Mgherony, Balázs Mikó

Institute of Material and Manufacturing Science, Óbuda University
Népszínház u. 8, H-1081 Budapest, Hungary
e-mail: mgherony.abdulwhab@uni-obuda.hu; miko.balazs@bgk.uni-obuda.hu

Abstract: Ball-end tools are widely used in industries such as die/mould, automobile, and the aerospace industry. These cutters are specially used when finishing complex three-dimensional sculptured surfaces. However, when using this cutter, the working diameter is different from the nominal diameter and varies from point to point due to the change in surface inclination. Therefore, cutting speed, which is calculated depending on the nominal diameter, is not accurate. The actual cutting speed changes resulting in several defects in the manufactured surface. This article introduces a new algorithm to improve the NC code of a CNC machine. The presented algorithm combines the STL format file with the APT language file to calculate the required spindle speed in order to provide a constant cutting speed during the entire machining process.

Keyword: Ball-end tool; surface roughness; cutting speed; STL format; APT language

Nomenclature

D_c	*tool diameter [mm]*
D_{eff}	*working diameter [mm]*
v_c	*cutting speed [m/min]*
n	*spindle speed [rpm]*
f_z	*feed per tooth [mm]*
z	*number of teeth [-]*
v_f	*feed rate [mm/min]*
A	*feed direction [°]*
a_p	*depth of cut [mm]*
a_e	*width of cut [mm]*
A_{N1}, A_{N2}	*surface inclination angles [rad]*
$\underline{P}_{t\,j}$	j^{th} *tool position point*
$\underline{P}_{tr\,i}$	i^{th} *triangle centre point*
$\underline{N}_{tr\,i}$	i^{th} *triangle normal vector*

Introduction

Free form surfaces have large importance in die and mould industry as a forming surfaces of sheet metal and plastic parts. The machining of forming insert is a piece production and generally done by milling technology or die sinking EDM technology, where the surface of the electrode machined by milling technology. The machining technology has to ensure the accurate surface geometry and fine and constant surface roughness. The surface roughness depends on many factors and circumstances, like the properties of the material, the type of machining, the geometric properties of the cutting tool, the cutting parameters, the tool path and the surface geometry.

Using ball-end tools for semi finishing and finishing free-form surfaces is common. The strong ability to adapt to different surface curvature gives this cutter an advantage over other cutting tools. Since the surface quality is not determined solely by cutting parameters, machining of the complex surfaces using ball-end tool still faces some challenges can be identified as follows: the drastic change in the geometrical features from roughing to finishing, the need to choose different strategies for the tool path, the performance of the CNC, the change in the cutting force and the change in the cutting speed. The current work focuses only on the effect of changing the cutting speed and proposes a method for controlling the spindle speed to ensure a constant cutting speed during the machining process.

During the ball-end milling process of sculpture surfaces, the working diameter of the tool changes. The working diameter value depends on the nominal diameter of the tool, the surface curvature, the feed direction, and the depth of cut. Some researcher studied the change in the working diameter, Mikó and Beňo [1] studied the effect of the inconstant effective diameter on the surface roughness, de Souza et al. [2] studied the effect of the working diameter in cutting force chip removal, and surface roughness, and found out that the surface roughness decreases by increasing the cutting speed. While Krajnik and Kopač [3] introduced a simple calculus of the working diameter when milling a surface by ball-end milling cutter and highlights the feedrate optimization in order to minimize the tool load. During the optimization, the G-code is modified directly.

Surface roughness quality and tool orientation were the focus of other studies. Sadílek et al. [4] compared the quality of the surface roughness in the case of 3-axis and 5-axis milling machines, they pointed out that small deviations of accuracy were deducted in the case of 5-axis milling. However, this deviation is higher in the case of 3-axis machine. Burek et al. [5] investigated the effect of changing tool direction on surface roughness and emphasized the importance of choosing lead and title angles. Gao et al. [6] investigated the impact of the machining inclination angles on the surface quality during machining Ti-6Al-4Vusing ball-end tool. The findings show that, the tool orientation planning has a major impact on the groove quality, and by choosing the appropriate inclination angle, the surface roughness can be reduced.

Many researchers studied the cutting speed and developed several approaches to guarantee a constant cutting speed when milling free-form surfaces. Varvruska et al. [7] proposed a method for milling using a circular cutting edge. In this method, the spindle speed and feed rate were controlled to achieve a constant cutting speed, and they found that under a constant cutting speed, the machining time is lower, and the productivity is higher compared to the traditional method. The working diameter is defined as the distance of the tool axis and the contact point of the surface and the tool. Zhang et al. [8] analysed the effect of the ball-end cutter speed and studied the relationships between the cutting speed and other important parameters such as feed direction and the angel of machining inclination. According to their results, the cutting speed has a significant effect on the machining process. Fan [9] presented in his work a mathematical model to calculate the actual cutting speed on the machined surface in 3-axis ball nose milling, based on the z-map method. Käsemodel et al. [10] presented an algorithm to control the spindle speed to keep the cutting speed constant. By using this algorithm, both the roughness and the machining time were reduced. This algorithm depends on the Grasshopper application at Rhino3D software to get the normal vector.

The aim of the research is to create a post-processing algorithm, which can modify the NC code to achieve spindle speed control and independence from CAD/CAM systems. The current article presents the general concept, the steps of the algorithm and the source of the required data. The geometric data of the surface can read from an STL file, and the tool path data can read from an APT format process file. The two standard file formats ensure the CAD/CAM independency. The work of the algorithm is demonstrated by an example.

1 Spindle Speed Control Concept

In order to the spindle speed control, the working diameter must be calculated point-by-point. Based on the applied geometric model [11], the calculation of the working diameter requires the description of the free form surface, the description of the cutting tool path and the cutting data.

In order to calculate the working diameter of the ball-end milling cutter, the surface date and the tool path data are required. The STL file format and the APT language were used to calculate the angle of the surface inclination at each cutting point, therefore, a brief description of the STL file and other file formats used to represent objects in CAD systems is given, followed by a short definition of the APT language and why it is used in this application.

1.1 STL File Format

To describe the geometric data of a free form surface, there are several possibilities from the native, CAD system-based file formats to the neutral, standard file formats. In the presented application, the method needs only the orientation of the surface normal vectors at different positions, so the different types of tessellation-based formats were analysed.

For rapid prototyping, CAD systems support many types of file formats such as STL, AMF, and OBJ. Each of these formats contain data such as texture, colour, and geometry data about the object to be manufactured. However, the data encoded in each of these types of formats is not the same and varies from file to file. For example, while an STL file only describes the surface geometry of an object without any representation of colour and texture, AMF and OBJ are capable to store information about texture and colour. Moreover, the difference between these formats is not only the type of information stored, but also the way the surface is represented. For example, STL uses planar triangles to describe the object, while AMF uses curved triangle in addition to planar straight triangles, OBJ can use more features such as free-form surfaces and free-form curves. Therefore, since STL file is the most used format and because only surface geometry is important apart from other information, STL has been used in this application [12] [13].

STL stands for Standard Triangle (or Tessellation) Language. This file format is widely used and supported by many CAD systems [14]. It consists of list of triangle facet data. Each facet includes facet normal and three vertices (Figure 1). The facet normal and the vertices are represented in three-dimensional Cartesian coordinate system. Below is an example of an STL file. The file begins with a "*SOLID*" and the name of the part, then a list of facets, and ends with an "*ENDSOLID*" followed by the name of the solid. The STL file includes only surface geometry data, while other information such as material, colour, and texture are not stored in this file format.

```
SOLID PRT0002
 facet normal 0.000e+00 0.000e+00 -1.000e+00
   outer loop
     vertex 0.000e+00 5.000e+01 0.000e+00
     vertex 5.000e+01 0.000e+00 0.000e+00
     vertex 0.000e+00 0.000e+00 0.000e+00
   endloop
 endfacet
...
ENDSOLID PRT0002
```

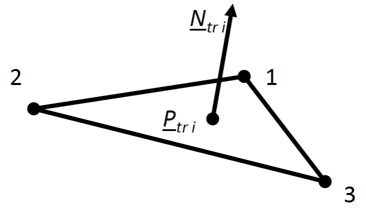

Figure 1
Definition of a triangle in the STL language

1.2 APT File

To get information about the tool path, the CNC code of the milling process should be read. Nowadays, G-code is the most common way to give instructions to numerical control machines. Although it is widely used, G-code is not that standard, every CNC controller has differences in syntax and how G-code is handled. Moreover, the location of the cutter cannot be obtained directly by reading the G-code required in this application. Therefore, the APT language was preferred to obtain the position of the tool during the milling process [15].

The Automated Programming Tool (APT) is a high-level computer programming language developed by the Massachusetts Institute of Technology (MIT) in 1958. It is used to automatically generate the program for the NC machines [16] [17]. Generally, all tool movement posture such as contact point, vector, cutting information and tool path are recorded in the APT. Through the data stored in the APT program, the NC code is generated [18].

As in the presented algorithm, there is a need to obtain the position of the cutter at each point. The structure of the APT language gives this possibility. Below is a part an APT program. The first section contains the main data of the program, like the part name (MFG0002), the machine tool ID (MILL, 1), the unit of the distances (MM), the tool ID (D10 R5) ant the direction of the coordinate system. The $$ marks comments. The next section contains the machining information, like the spindle speed (SPINDL), and the points of the tool motion. The RAPID marks the fast motion, and after the FEDRAT the working motion starts. The GOTO indicates the x,y and z coordinates of the end point of a linear motion.

```
PARTNO / MFG0002
MACHIN / MILL, 1
UNITS / MM
LOADTL / 37 $$-> D10 R5
$$-> CUTTER / 10.000000
$$-> CSYS / 1.000, 0.000, 0.000, -25.000,  $
            0.000, 1.000, 0.000, -25.000,  $
            0.000, 0.000, 1.000, 0.000
SPINDL / RPM, 2000.00, CLW
RAPID
FROM / -20.100, 25.249, 30.000
RAPID
GOTO / -20.100, 25.249, 9.994
FEDRAT / 500.00, MMPM
GOTO / -19.975, 25.249, 9.994
GOTO / -18.996, 25.249, 9.990
…
```

As can be seen from the example above, the APT program starts with some information about the part, machine and unit of measure followed by cutting parameters such as tool diameter and spindle speed. Then, there is a list of tool positions. The x, y and z coordinates of the tool position are marked in every case, not only in changing parameters, like in G coding. This is one of the advantages of the APT description of the tool path. In this example, the process is down milling, so not all the movements are cutting movements. However, the algorithm is capable to distinguish between cutting motions and rapid motions as the location of the cutter is only important when there is contact between the tool and the workpiece.

The other data of the process, such as the diameter of the cutting tool, the nominal cutting speed and nominal feed rate, the depth of cut, the width of cut, and the milling direction can be determined based on the APT file, or they can be user inputs.

1.3 The Algorithm Description

To achieve the aim of this study, an algorithm was developed to control and adjust the spindle speed point-by-point to keep the cutting speed constant. Figure 2 illustrates how this algorithm works to calculate the spindle speed required at each point of the surface.

The algorithm starts by reading the STL file. The STL file is created by the CAD system to describe the surface of a workpiece. The algorithm reads the file and calculates the incentre point for each triangle by the calculation of the average values of the coordinates of the three corner of a triangle. At the end of this stage, the algorithm will store a list of incentre points ($\underline{P}_{tr\ i}$) of the triangles with the corresponding normal vectors ($\underline{N}_{tr\ i}$). For a more efficient work, the triangles, which are not the part of the machined surface, are eliminated. In order to elimination, the normal vector of a triangle is investigated. If the normal vector is parallel with a coordinate axes, except of $Z+$ direction, the triangle is skipped.

Then, the APT file is read by the algorithm. In the current research linear interpolation was used during the generation of the APT file. The APT file contains the coordinates of the tool location. However, these points are not the contact point between the tool and the surface, but they can use for the calculation. The difference between the tool nose point and the tool/surface contact point is small, especially in case of shallow surfaces. The machining points are listed after the "*FEDRAT*" command, while the "*RAPID*" marks the points of the fast motion of the tool. So the different types of the tool location can be separated by the investigation of these key words.

The next step is to find the normal vector at each position of the tool. This can be done by connecting the position of the tool to the corresponding triangle. Therefore, the algorithm calculates the distance between the tool location and all the incentre points and returns the incenter point with the shortest distance to the tool position. That means, the tool is in the triangle which has that incenter point and it has the same normal vector as the triangle.

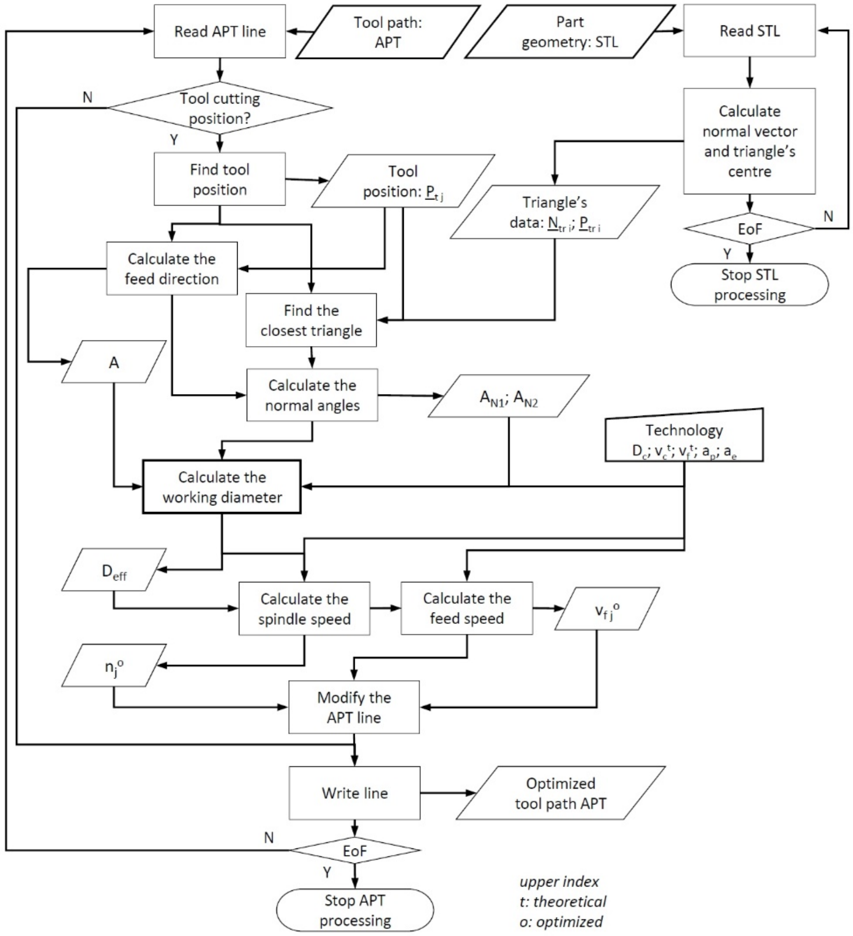
Read APT line
Tool path: APT
Part geometry: STL
Read STL
Tool cutting position?
N
Y
Calculate normal vector and triangle's centre
Find tool position
Tool position: $\underline{P}_{t\,j}$
Triangle's data: $\underline{N}_{tr\,i}$; $\underline{P}_{tr\,i}$
EoF
N
Y
Stop STL processing
Calculate the feed direction
Find the closest triangle
A
Calculate the normal angles
A_{N1}; A_{N2}
Technology D_c; v_c^t; v_f^t; a_p; a_e
Calculate the working diameter
D_{eff}
Calculate the spindle speed
Calculate the feed speed
$v_{f\,j}^o$
n_j^o
Modify the APT line
Write line
Optimized tool path APT
N
EoF
Y
Stop APT processing
upper index
t: theoretical
o: optimized

Figure 2
Chart of the algorithm

Depending on the resolution of the STL file, one triangle can be assigned to more tool position. It can be a source of the error in the calculation of the working diameter, but it has just a small effect, and the accuracy can be improved by the resolution of the surface description.

Through finding the normal vector of the surface at each tool position, the surface inclination angles AN_1 and AN_2 can be calculated based on the Figure 3.

$$A_{\mathrm{N1}} = arctg\left(\frac{N_y}{N_x}\right) \tag{1}$$

$$A_{\mathrm{N2}} = arccos(N_z) \tag{2}$$

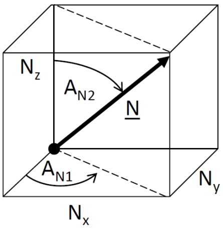

Figure 3
Calculation of the inclination angles (A_{N1}, A_{N2})

If the cutter location is known, the feed direction can be calculated as well (Fig. 4).

$$A = arctg\left(\frac{P_{y\_(j)}-P_{y\_(j-1)}}{P_{x\_(j)}-P_{x\_(j-1)}}\right) \tag{3}$$

The feed direction can be constant, like in the presented example. In this case, there is no need to calculate it point-by-point, it can be an input data.

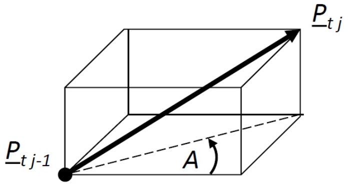

Figure 4
Calculation of the feed direction (A)

These data are necessary to calculate the working diameter at each tool position, which is an important step to calculate the required spindle speed. To calculate the working diameter, a geometric model presented by Mikó and Zentay [11] was used.

In this model, the working diameter is defined as a distance of the tool centre line and an intersection circle. The intersection circle is generated by plane, which has the same normal, as the current surface point and the distance is equal with the depth of cut. The required transformation is driven by the inclination angles A_{N1} and A_{N2}. The direction of the working diameter is defined by the feed direction. The machined part geometry of the free form surface is approximated locally in this model, which harmonizes with the use of the STL description of the surface geometry.

In order to ensure the constant cutting speed, the spindle speed has to be calculated point-by-point. The required spindle speed at each point is calculated by the following formula:

$$n_o = \frac{1000*v_c}{D_{eff}*\pi} \text{ [1/min]} \tag{4}$$

For constant cutting condition, the feed rate has to be modified parallel with the spindle speed:

$$v_f = f_z * z * n \text{ [mm/min]} \tag{5}$$

2 The Results and the Simulation

A simulation was carried out to study the change in the cutting speed and the required change in the spindle speed to compromise the cutting speed in order to obtain a constant cutting speed using the presented algorithm. Table 1 shows the cutting parameters which were used in this simulation. While Figure 5 shows a simulation of the down milling process in CAM software.

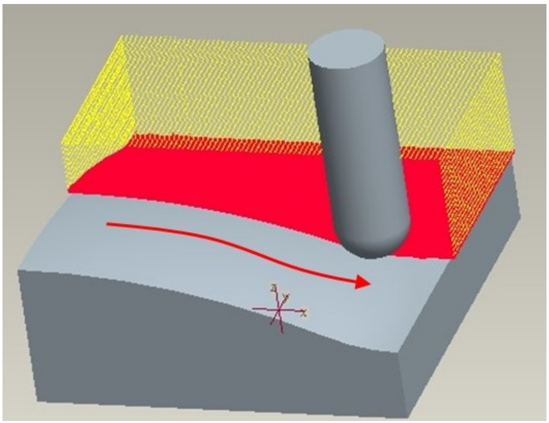

Figure 5
Down milling process in CAM simulation

Table 1
Cutting parameters

Cutting speed	v_c [m/min]	63
Spindle speed	n [rpm]	2000
Feed per tooth	f_z [mm]	0.125
Feed rate	v_f [mm/min]	500
Feed direction	A [°]	0
Depth of cut	a_p [mm]	0.30
Width of cut	a_e [mm]	0.25
Tool diameter	D [mm]	10.0
Number of teeth	z [-]	2

The STL file describes a workpiece with 50 x 50 mm (Figure 6). While the APT file contains the instructions for a down milling process, by reading this file the tool location can be determined by the algorithm as it can be seen in Figure 7. However, the resolution of the STL file is less than the resolution of the tool path description, this error is not significant from the viewpoint of the spindle speed. The coordinates of the tool position are different from the coordinates of the surface points because the tool control point is located on the nose point of the tool. This difference is not so large, than it affects the optimization goals.

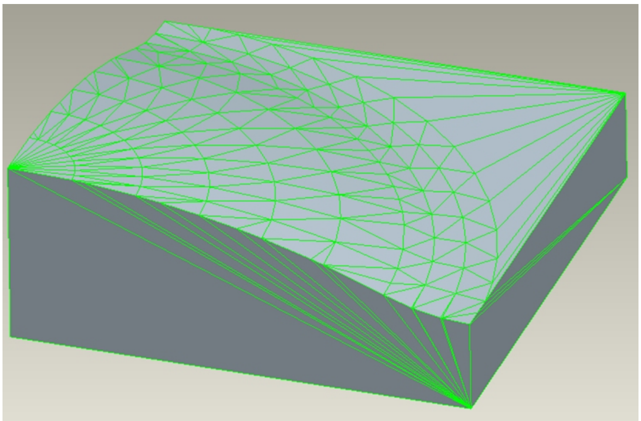

Figure 6

The STL representation of the test surface

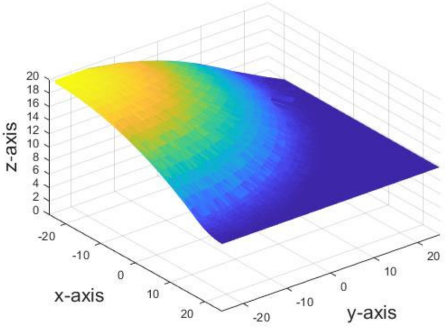

Figure 7

The tool position determined by reading the APT file

The working diameter and cutting speed were calculated in the case of feed direction *A =0*. Figure 8 and Figure 9 show the change in the working diameter and cutting speed from point to point along the surface of the workpiece. The working diameter of the tool and the actual cutting speed change parallel of course. When the surface normal close to vertical direction (the surface is horizontal), the D_{eff} is small and the cutting speed decreases.

As can be seen in Figure 8, the working diameter drops to the lowest value (about 1.5 mm) when the surface is horizontal, in this section of the workpiece, the axis of the tool is normal on the surface. While in the middle, the diameter of the work increases as the tool engages more into the workpiece. The actual cutting speed changes from point to point along the surface, similar to the working diameter.

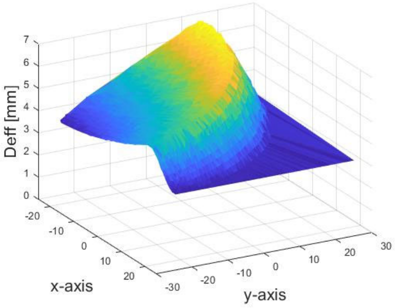

Figure 8

The working diameter

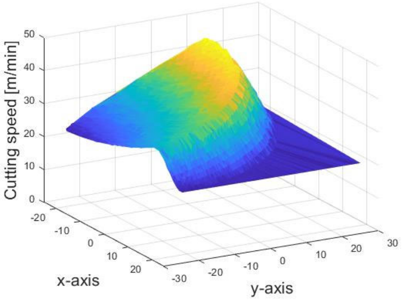

Figure 9

The cutting speed

Similarly, Figure 10 and Figure 11 show the working diameter and cutting speed as a function of the surface slope. When $A_{N1} = 0$ and $A_{N2} = 0$, the working diameter is of low value. The higher the surface slop, the greater the working diameter and more work done by the cutter. This also applies to cutting speed.

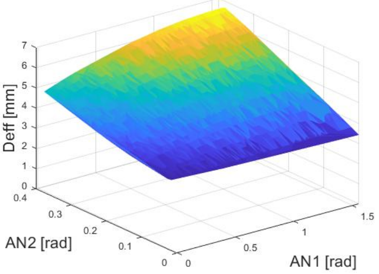

Figure 10

Working diameter versus surface inclination

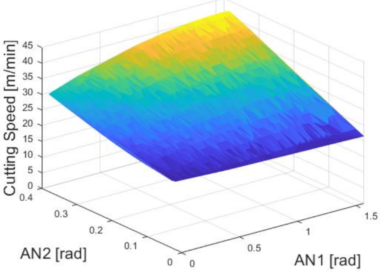

Figure 11

Cutting speed versus surface inclination

Although the cutting parameter including spindle speed is constant, the actual cutting speed changes from point to point during the milling process as a result of the change in tool working diameter which varies with different surface inclination angles. Even though, the nominal cutting speed is 63 [m/min], the actual cutting speed is much lower, and the highest speed can reach 42 [m/min], at some points, while decreases to about 21 [m/min] in other cases.

To keep the cutting speed constant, the spindle speed must be changed and in order to keep the feed per tooth value, the feed rate has to be modified too. The presented algorithm gives a solution to the problem of controlling the spindle speed as well as the feed rate. Figure 12 and Figure 13 show the modified spindle speed and feed rate at each point to keep the cutting speed constant at 63 [m/min].

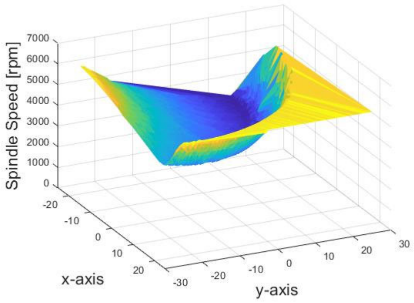

Figure 12
Required spindle speed

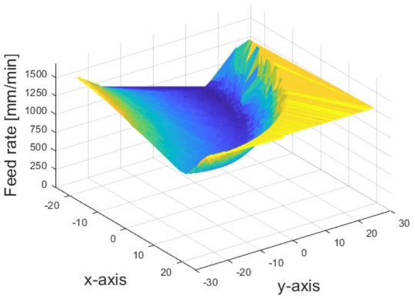

Figure 13
Required feed rate

At some points, an increase in the spindle speed is required to compensate for the decrease in cutting speed that occurs as a result of a decrease in the value of the working diameter. The spindle speed required at these points can reach 5877 [rpm] which is three times the present spindle speed. The lowest spindle speed is 3002 [rpm]. On the other hand, to achieve the same purpose of keeping the cutting speed constant, the feed rate should also be adjusted. The increase in the required feed rate is directly proportional to the spindle speed. Therefore, both parameters have the same character, and in a similar way, the increase in the feed rate reaches 1500 [mm/min], which is three times greater than the nominal feed rate. The feed rate is changed between 750 and 1469 mm/min.

By the mean of the presented algorithm, the cutting speed can be fixed to a specific value. Figure 14 shows the theoretical cutting speed as a function of surface slope. The cutting speed maintains to a constant value (63 [m/min]) regardless of the change of surface angles. In case of the simulation, a small difference can occur only in the cutting speed depending on the resolution of the calculation. The resolution is dependent on the density of the tool path points and the STL surface model.

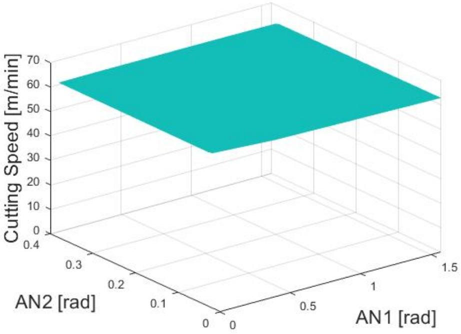

Figure 14
Theoretical cutting speed

Conclusion

The change in the working diameter of the milling cutter is the main problem faced in the manufacture of free-form surfaces using a three-axis ball end mill. Since the working diameter varies from point to point along the surface, the cutting speed also varies, which affects the cutting parameters and as a result the surface quality.

In this article, an algorithm is presented to control the spindle speed to obtain a constant cutting speed. In contrast, to the other approach to adjusting the spindle speed, the presented algorithm does not need any pre-processing and it is independent from any CAD or CAM system. The algorithm uses a new idea to get the normal vector at each point of the surface, which includes reading the STL format file and the NC file written in APT language and making a connection between the cutter locations and the triangles of the STL file. The developed algorithm based on the surface normal from STL and the feed direction from APT can determine the working diameter of the tool, and the spindle speed can be calculated in order to ensure the constant cutting speed.

A simulation was used to demonstrate how the algorithm works. The simulation shows how the working diameter and the cutting speed change during the milling process, due to the change in the angles of the surface slop. Besides, it illustrates how the spindle speed and feed rate should be adjusted to ensure a constant cutting speed through the entire milling process. The presented algorithm can implement inside a CAM system, or it can also work as an independent optimization application.

During future work the experiments have to demonstrate the effect of the constant cutting speed to the surface quality. These experiments include machine workpieces with different feeding directions in the case of constant spindle speed and in the case of constant cutting speed, measure the surface roughness in both cases and compare the results.

As the results show, there can be a large change in the spindle speed. So on one hand, the dynamic properties of the milling machine have to be investigated. The execution of the modified machining program can generate high and dynamic load of the spindle. Based on this result, the algorithm should be modified in order to decrease this dynamic load. On the other hand, the change of the spindle speed can decrease by a new type of the surface milling strategy, which can be developed based on the calculation of the working diameter of the ball-end milling cutter. The next step is to design a tool path searching algorithm, which can generate a new tool path based on the previous one, which considers the geometric information of the surface from the STL file and minimize the changing of the spindle speed.

References

[1] B. Mikó, J. Beňo: Effect of the working diameter to the surface quality in free-form surface milling. Key Engineering Materials 581:372-377 (2014) doi: 10.4028/www.scientific.net/KEM.581.372

[2] A. F. de Souza, A. E. Diniz, A. R. Rodrigues, R. T. Coelho: Investigating the cutting phenomena in free-form milling using a ball-end cutting tool for die and mold manufacturing. The International Journal of Advanced Manufacturing Technology 71(9):1565-1577 (2014) doi: 10.1007/s00170-013-5579-4

[3] P. Krajnik, J. Kopač: Modern machining of die and mold tools. Journal of Materials Processing Technology 157:543-552 (2004) doi: 10.1016/j.jmatprotec.2004.07.146

[4] M. Sadílek, L. Kousal, N. Náprstková, T. Szotkowski, J. Hajnyš: The analysis of accuracy of machined surfaces and surfaces roughness after 3axis and 5axis milling. Manufacturing Technology 18(6):1015-1022 (2018) doi: 10.21062/ujep/217.2018/a/1213-2489/MT/18/6/1015

[5] J. Burek, T. Rydzak, A. Szajna: The influence of the tool axis orientation changes and the direction of tool driving in free surface machining on surface roughness. Mechanik 91(8-9):684-686 (2018) doi: 10.17814/mechanik.2018.8-9.107

[6] P. Gao, X. Wang, Z. Liang, S. Zhang, T. Zhou, P. Yan, L. Jiao: Effects of machining inclination angles on microgroove quality in micro ball end milling of ti-6al-4v. The International Journal of Advanced Manufacturing Technology 92(5):2725-2734 (2017) doi: 10.1007/s00170-017-0305-2

[7] P. Vavruska, P. Zeman, M. Stejskal: Reducing machining time by pre-process control of spindle speed and feed-rate in milling strategies. Procedia CIRP 77:578-581 (2018) doi: 10.1016/j.procir.2018.08.216

[8] A. Zhang, X. Liu, C. Yue, R. Li, S. Y. Liang, L. Wang: Velocity effect sensitivity analysis of ball end milling Ti-6Al-4V. The International Journal of Advanced Manufacturing Technology 118:3963-3982 (2022) doi: 10.1007/s00170-021-08049-4

[9] J. Fan: Cutting speed modelling in ball nose milling applications. The International Journal of Advanced Manufacturing Technology 73(1):161-171 (2014) doi: 10.1007/s00170-014-5672-3

[10] R. B. Käsemodel, A. F. de Souza, R. Voigt, I. Basso, A. R. Rodrigues: CAD/CAM interfaced algorithm reduces cutting force, roughness, and machining time in free-form milling. The International Journal of Advanced Manufacturing Technology 107(3):1883-1900 (2020) doi: 10.1007/s00170-020-05143-x

[11] B. Mikó, P. Zentay: A geometric approach of working tool diameter in 3-axis ball-end milling. The International Journal of Advanced Manufacturing Technology 104(1):1497-1507 (2019) doi: 10.1007/s00170-019-03968-9

[12] C. Iancu: About 3d printing file formats. Annals of the Constantin Brancusi University of Targu Jiu-Letters & Social Sciences Series 2:135-138 (2018)

[13] M. Szilvási-Nagy, Gy. Mátyasi: Analysis of STL files. Mathematical and Computer Modelling 38(7-9):945-960 (2003) doi: 10.1016/S0895-7177(03)90079-3

[14] Y. Chen, C. Ng, Y. Wang: Generation of an STL file from 3d measurement data with user-controlled data reduction. The International Journal of Advanced Manufacturing Technology 15(2):127-131 (1999) doi: 10.1007/s001700050049

[15] S. Brown, C. Drayton, B. Mittman: A description of the APT language. Communications of the ACM 6(11):649-658 (1963) doi: 10.1145/368310.368322

[16] H. Y. Cheng, J. P. Tsai, Y. C. Kao: The development of an apt program interpreter for 5-axis machining. Advanced Materials Research 482:2247-2252 (2012) doi: 10.4028/www.scientific.net/AMR.482-484.2247

[17] D. T. Ross: The design and use of the apt language for automatic programming of numerically controlled machine tools. Proc. 1959 Computer Applications Symposium, Chiergo, pp. 80-99 (1959) doi: 10.1145/960118.808374

[18] D. T. Ross: Origins of the apt language for automatically programmed tools. History of Programming Languages June:279-338 (1978)